Logistic discrimination with total variation regularization

Robin Rühlicke
University of Ulm

Daniel Gervini *
University of Wisconsin–Milwaukee

September 9, 2007

Abstract

This article introduces a regularized logistic discrimination method that is especially suited for discretized stochastic processes (such as periodograms, spectrograms, EEG curves, etc.). The proposed method penalizes the total variation of the discriminant directions, giving smaller misclassification errors than alternative methods, and smoother and more easily interpretable discriminant directions. The properties of the new method are studied by simulation and by a real-data example involving classification of phonemes.

*Supported in part by NSF Grant DMS-06-04396.
Key words and phrases: Classification, discrimination, machine learning, speech recognition.

1 Introduction

The problem of classifying data into predefined classes is encountered in diverse fields of applications, such as speech recognition and image analysis. Figure 1a shows a typical example: one of the curves is a log-periodogram corresponding to the phoneme 'aa' (like the 'a' in 'dark') and the other one is a log-periodogram corresponding to the phoneme 'ao' (like the 'a' in 'water'); the task is to identify which phoneme was pronounced.

Mathematically, we can describe the classification problem as follows: let $\boldsymbol{X} \in \mathbb{R}^m$ be a vector of features and G a group-indicator variable; we assume that each object can be classified into one and only one of K predefined classes, so $G \in \mathcal{G} = \{1, \ldots, K\}$. Given a training sample $(\boldsymbol{x}_1, g_1), \ldots, (\boldsymbol{x}_n, g_n)$ of objects with known classification, the goal is to construct a classification rule $\hat{G} : \mathbb{R}^m \to \mathcal{G}$ for future observations. Many techniques for supervised classification have been proposed over the years; an up-to-date overview is presented in Hastie, Tibshirani and Friedman (2001).

One of these methods is logistic discrimination. It assumes that the log-odds of the conditional distribution of G given $\boldsymbol{X} = \boldsymbol{x}$ are linear functions of $\boldsymbol{x}$:

$$\log\left\{\frac{p_{G|\boldsymbol{X}}(k|\boldsymbol{x})}{p_{G|\boldsymbol{X}}(K|\boldsymbol{x})}\right\} = \alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}, \quad k = 1, \ldots, K-1. \tag{1}$$

Then

$$p_{G|\boldsymbol{X}}(k|\boldsymbol{x}) = \frac{\exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})}{1 + \sum_{l=1}^{K-1} \exp(\alpha_l + \boldsymbol{\beta}_l^T \boldsymbol{x})}, \quad k = 1, \ldots, K-1, \tag{2}$$

and

$$p_{G|\boldsymbol{X}}(K|\boldsymbol{x}) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\alpha_l + \boldsymbol{\beta}_l^T \boldsymbol{x})}. \tag{3}$$

The parameters $\{(\alpha_k, \boldsymbol{\beta}_k)\}$ are usually estimated by conditional maximum likeli-

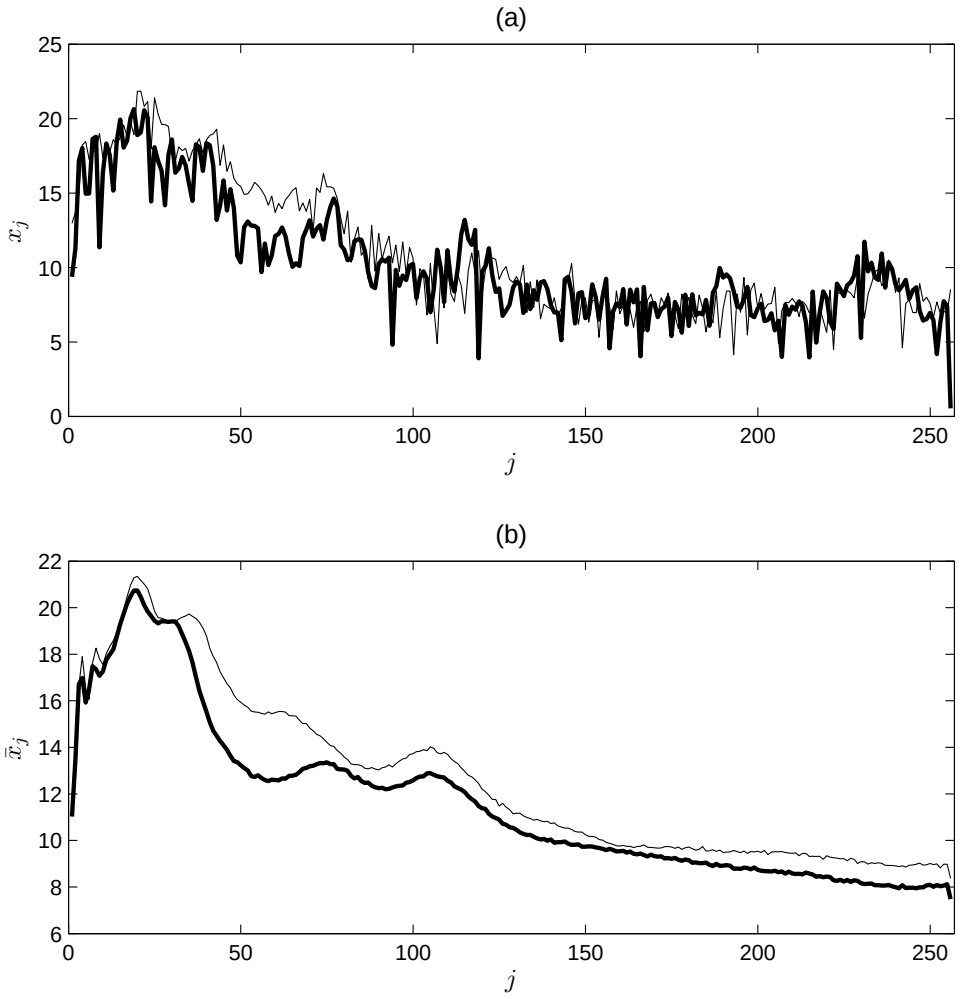

Figure 1: Phoneme recognition example. Log-periodograms of the phonemes 'aa' (thin line) and 'ao' (thick line); (a) shows two sample curves and (b) the sample means.

hood; that is, by maximizing

$$L(\alpha_1, \ldots, \alpha_{K-1}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}) = \sum_{i=1}^{n} \log p_{G|\boldsymbol{X}}(g_i|\boldsymbol{x}_i). \tag{4}$$

Logistic discrimination has many appealing properties. One of them is robustness: it handles outliers and qualitative variables better than Fisher's linear discrimination, which is the optimal procedure under Gaussian assumptions (Krzanowski (1977)).

Difficulties arise when the dimension of $\boldsymbol{X}$ is large compared to the sample size. In such cases, the maximizers $\{\hat{\boldsymbol{\beta}}_k\}$ of (4) are very irregular, and while they provide good fits for the training data, they yield poor classification rates for new observations. This problem can be alleviated by regularizing the estimators, which is usually done by maximizing a penalized version of (4). For instance, Hastie et al. (2001) suggest maximizing

$$L(\alpha_1, \ldots, \alpha_{K-1}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}) - \lambda \sum_{k=1}^{K-1} \|\boldsymbol{\beta}_k\|_2^2, \tag{5}$$

where $\|\boldsymbol{\beta}\|_2$ is the Euclidean norm and λ is a non-negative penalization parameter. So far this is the only kind of regularized logistic discrimination that has been studied in the literature (Hastie et al. (2001), Zhu and Hastie (2004)).

However, other forms of regularization are possible. For instance, it is becoming increasingly popular to use the L^1 norm $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{m} |\beta_j|$ for penalization. The advantage of the L^1 norm over the L^2 norm (at least in the context of linear regression) is that it tends to produce sparse estimators with many coefficients equal to zero, which represents a kind of automatic variable selection (see e.g. Tibshirani (1996), Fan and Li (2001), and Fan and Peng (2004)).

In certain situations, however, it is of no practical use that a few isolated coordinates of $\hat{\boldsymbol{\beta}}$ be equal to zero; it is more useful that *consecutive* coordinates of $\hat{\boldsymbol{\beta}}$ be equal to zero. For example, we see in Fig. 1 that only a narrow range of frequencies are needed to discriminate 'aa' from 'ao', the rest being mostly random noise; so it

is reasonable to expect that $\beta_j = 0$ for $j \geq 100$, say. Situations like this are common when the feature vector $\boldsymbol{X}$ is the discretization of a continuous-time stochastic process, of which there are many examples in applications: periodograms, spectrograms, EEG signals, and gene expression profiles, to name a few.

To obtain estimators $\hat{\boldsymbol{\beta}}$ with constant consecutive coordinates, the *difference* between consecutive coordinates of $\boldsymbol{\beta}$ should be penalized, rather than the coefficients β_j themselves. From a computational point of view, the simplest estimators of this kind would be the maximizers of

$$L(\alpha_1, \ldots, \alpha_{K-1}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}) - \lambda \sum_{k=1}^{K-1} \{\beta_{k1}^2 + \sum_{j=2}^{m} (\beta_{kj} - \beta_{k,j-1})^2\}. \tag{6}$$

However, an alternative that will turn out to work better in practice (even if it is computationally more complex) is to maximize

$$L(\alpha_1, \ldots, \alpha_{K-1}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}) - \lambda \sum_{k=1}^{K-1} \{|\beta_{k1}| + \sum_{j=2}^{m} |\beta_{kj} - \beta_{k,j-1}|\}. \tag{7}$$

Note that (7) penalizes the *total variation* of the $\hat{\boldsymbol{\beta}}_k$s; that is, the absolute value of the difference between consecutive coefficients.

In this article we show, by example and simulations, that the estimators defined as the maximizers of (7) are able to attain lower misclassification errors than the other regularized estimators. Moreover, the $\hat{\boldsymbol{\beta}}_k$s obtained from (7) are less variable and easier to interpret than those obtained with alternative methods. Their computation, however, is more complicated, so we devote the next two sections to computational issues.

2 Estimating equations

For mathematical convenience, let $\boldsymbol{y}_i \in \mathbb{R}^K$ be indicator vectors with $y_{ik} = 1$ if $g_i = k$ and $y_{ik} = 0$ otherwise. Then the log-likelihood function (4) simplifies to

$$\begin{aligned} & L(\alpha_1, \ldots, \alpha_{K-1}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log p_{G|\boldsymbol{X}}(k|\boldsymbol{x}_i) \\ &= \sum_{i=1}^{n} \left[\sum_{k=1}^{K-1} y_{ik} \log \left\{ \frac{p_{G|\boldsymbol{X}}(k|\boldsymbol{x}_i)}{p_{G|\boldsymbol{X}}(K|\boldsymbol{x}_i)} \right\} + \log p_{G|\boldsymbol{X}}(K|\boldsymbol{x}_i) \right] \\ &= \sum_{i=1}^{n} \left[\sum_{k=1}^{K-1} y_{ik}(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i) - \log\{1 + \sum_{k=1}^{K-1} \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i)\} \right]. \end{aligned}$$

The total-variation regularized estimators, defined as the maximizers of (7), always exist if $\lambda > 0$, since L is concave and non-positive and the penalization term rules out the possibility of approaching the maximum at infinity. So the first-order estimating equations that we derive next will always have solutions.

For computational purposes, it is convenient to reparameterize (7) as follows: let $\boldsymbol{\gamma}_k = (\beta_{k1}, \beta_{k2} - \beta_{k1}, \ldots, \beta_{km} - \beta_{k,m-1})^T$, and $\boldsymbol{z}_i = (A^{-1})^T \boldsymbol{x}_i$, where A is the transformation matrix such that $\boldsymbol{\gamma}_k = A\boldsymbol{\beta}_k$. Then, maximizing (7) is equivalent to minimizing

$$\sum_{i=1}^{n} \left[\sum_{k=1}^{K-1} y_{ik}(\alpha_k + \boldsymbol{\gamma}_k^T \boldsymbol{z}_i) - \log\{1 + \sum_{k=1}^{K-1} \exp(\alpha_k + \boldsymbol{\gamma}_k^T \boldsymbol{z}_i)\} \right] + \lambda \sum_{k=1}^{K-1} \|\boldsymbol{\gamma}_k\|_1. \qquad (8)$$

First-order estimating equations for $\{(\hat{\alpha}_k, \hat{\boldsymbol{\gamma}}_k)\}$ must be derived with care, due to the non-differentiability of the absolute value function at zero. However, a simple convexity argument shows that the partial derivative of $-L$ with respect to γ_{kj} has

to be zero if $\hat{\gamma}_{kj} = 0$, so the estimating equations are:

$$\begin{aligned} -\sum_{i=1}^{n}\left\{ y_{ik} - \frac{\exp(\alpha_k + \boldsymbol{\gamma}_k^T \boldsymbol{z}_i)}{1 + \sum_{l=1}^{K-1} \exp(\alpha_l + \boldsymbol{\gamma}_l^T \boldsymbol{z}_i)} \right\} &= 0, \\ -\sum_{i=1}^{n}\left\{ y_{ik} - \frac{\exp(\alpha_k + \boldsymbol{\gamma}_k^T \boldsymbol{z}_i)}{1 + \sum_{l=1}^{K-1} \exp(\alpha_l + \boldsymbol{\gamma}_l^T \boldsymbol{z}_i)} \right\} \boldsymbol{z}_i + \lambda \, \mathrm{sign}(\boldsymbol{\gamma}_k) &= \mathbf{0}, \end{aligned}$$

for $k = 1, \ldots, K-1$. Here the sign function is understood in a componentwise manner, and $\mathrm{sign}(0) = 0$.

Estimators of the $\boldsymbol{\beta}_k$s are obtained by back-transforming the $\hat{\boldsymbol{\gamma}}_k$s; explicitly, $\hat{\boldsymbol{\beta}}_k = A^{-1}\hat{\boldsymbol{\gamma}}_k$. The classification rule $\hat{G}$ is then

$$\hat{G}(\boldsymbol{x}) = \{k : \hat{p}_{G|\boldsymbol{X}}(k|\boldsymbol{x}) \geq \hat{p}_{G|\boldsymbol{X}}(j|\boldsymbol{x}),\ j = 1, \ldots, K\},$$

where $\{\hat{p}_{G|\boldsymbol{X}}(k|\boldsymbol{x})\}$ are as in (2) and (3), only that the true parameters are replaced by their estimators.

3 Algorithm for the two-group case

The example and simulations presented in sections 4 and 5 are two-group discrimination problems, so we now explain in detail the algorithm we have used for that case. Let $\boldsymbol{\theta} = (\alpha_1, \boldsymbol{\beta}_1^T)^T$, $\boldsymbol{w}_i = (1, \boldsymbol{z}_i^T)^T$, and define the indicator variables $y_i \in \{0, 1\}$ as $y_i = 1$ if $g_i = 1$ and $y_i = 0$ if $g_i = 2$. Then (8) becomes

$$F(\boldsymbol{\theta}) = -\sum_{i=1}^{n} y_i \log \pi(\boldsymbol{w}_i^T \boldsymbol{\theta}) - \sum_{i=1}^{n} (1 - y_i) \log \{1 - \pi(\boldsymbol{w}_i^T \boldsymbol{\theta})\} + \lambda \sum_{j=2}^{m} |\theta_j|,$$

where $\pi(t)$ is the logistic function. The gradient (or, more precisely, the subgradient) of $F(\boldsymbol{\theta})$ is

$$\nabla F(\boldsymbol{\theta}) = -W\{\boldsymbol{y} - \pi(W\boldsymbol{\theta})\} + \lambda \, [0, \mathrm{sign}(\boldsymbol{\gamma})^T]^T,$$

where W is a matrix with the ith row equal to $\boldsymbol{w}_i^T$.

To find the minimizer of $F(\boldsymbol{\theta})$ we use a nonlinear conjugate gradient method based on Algorithm 5.4 of Nocedal and Wright (1999). Our algorithm initially takes a step in the steepest descent direction, performing a line search to find the minimum of the objective function in that direction. Subsequent directions are computed as linear combinations of the previous directions (using the Polak-Ribière method), reverting back to steepest descent when two consecutive gradients are too far from orthogonal. As starting value for $\boldsymbol{\theta}$ we take $\boldsymbol{\theta}_0 = \mathbf{0}$. For the other parameters of Algorithm 1 we use $max\_err = 10^{-6}$, $max\_iterations = 5000$ and $c_t = 1$.

Algorithm 1

Given $\boldsymbol{\theta}_0, max\_err, max\_iterations, c_t$
$F_0 = F(\boldsymbol{\theta}_0)$ and $\nabla F_0 = \nabla F(\boldsymbol{\theta}_0)$
$\boldsymbol{p}_0 = -\nabla F_0$
$k = 0$
$err = 1$
while $err > max\_err$ and $k < max\_iterations$ **do**
  Compute α_k with line search algorithm
  $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k \boldsymbol{p}_k$
  $F_{k+1} = F(\boldsymbol{\theta}_{k+1})$
  $\nabla F_{k+1} = \nabla F(\boldsymbol{\theta}_{k+1})$
  if $\frac{\nabla F_{k+1}^T \nabla F_k}{\nabla F_{k+1}^T \nabla F_{k+1}} > c_t$ **then**
    $\beta_{k+1}^{PR} = 0$
  else
    $\beta_{k+1}^{PR} = max\{0, \frac{\nabla F_{k+1}^T (\nabla F_{k+1} - \nabla F_k)}{\nabla F_k^T \nabla F_k}\}$
  end if
  $p_{k+1} = -\nabla F_{k+1} + \beta_{k+1}^{PR} \boldsymbol{p}_k$
  if $\|\boldsymbol{\theta}_{k-1}\| > 0$ **then**
    $err = \|\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_k\| / \|\boldsymbol{\theta}_{k-1}\|$
  else
    $err = \|\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_k\|$
  end if
  $k = k + 1$
end while

A crucial part of the algorithm is the line search step. The line search starts with a guess for the steplength α_0 and tries to minimize F along the descent direction $\boldsymbol{p}_k$, $\Phi(\alpha) = F(\boldsymbol{\theta} + \alpha \boldsymbol{p}_k)$, with respect to α. If the initial guess α_0 yields a sufficient decrease of the objective function, i.e., if it fulfills the strong Wolfe conditions

$$\Phi(\alpha_0) \leq \Phi(0) + c_1 \alpha_0 \Phi'(0), \tag{9}$$

$$|\Phi'(\alpha_0)| \leq c_2 |\Phi'(0)|, \tag{10}$$

for some constants $0 < c_1 < c_2 < \frac{1}{2}$ (we take $c_1 = 0.0001$ and $c_2 = 0.4$), then α_0 is used as steplength. Otherwise, we approximate Φ by quadratic polynomial interpolation of the points $(0, \Phi(0)) = (0, F_k)$, $(0, \Phi'(0)) = (0, F_k^T \boldsymbol{p}_k)$ and $(\alpha_0, \Phi(\alpha_0))$, and take the minimizer of this polynomial as step length (Nocedal and Wright (1999) prove that the interpolating polynomial always has a minimum). Explicitly, this is

$$\alpha_1 = \frac{\alpha_0^2 \Phi'(0)}{2[\Phi(\alpha_0) - \alpha_0 \Phi'(0) - \Phi(0)]}.$$

If α_1 does not satisfy conditions (9) and (10), Φ is approximated by cubic interpolation of the points $(0, \Phi(0))$, $(0, \Phi'(0))$, $(\alpha_{j-1}, \Phi(\alpha_{j-1}))$ and $(\alpha_j, \Phi(\alpha_j))$, with $j = 1$ in this case. The minimizer of the cubic interpolant is

$$\alpha_{j+1} = \frac{-b + \sqrt{b^2 - 3a\Phi'(0)}}{3a},$$

where

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_j^3 \alpha_{j-1}^2 - \alpha_{j-1}^3 \alpha_j^2} \begin{bmatrix} \alpha_{j-1}^2 & -\alpha_j^2 \\ -\alpha_{j-1}^3 & \alpha_j^3 \end{bmatrix} \begin{bmatrix} \Phi(\alpha_j) - \Phi'(0)\alpha_j - \Phi(0) \\ \Phi(\alpha_{j-1}) - \Phi'(0)\alpha_{j-1} - \Phi(0) \end{bmatrix}.$$

Note that $\alpha_{j+1} \in [0, \alpha_j]$ (see Nocedal and Wright (1999) for a proof). The cubic interpolation step is iterated until a steplength α_{j+1} that fulfills (9) and (10) is found. Otherwise the algorithm terminates after a given number of iterations (we use 50), or if α_{j+1} is either too close to 0 or too close to α_j. A Matlab program implementing these algorithms is available at the second author's website,

Table 1: Phoneme recognition example. Misclassification errors.

			LDQD		LDTV	
Source	LD	LDQC	CV opt	True opt	CV opt	True opt
Training set	0.093	0.142	0.152	0.165	0.164	0.174
Test set	0.244	0.187	0.196	0.187	0.189	0.182

http://www.uwm.edu/~gervini.

4 Example: Phoneme Recognition

In this section we study the performance of four different classification methods on the phoneme recognition problem studied by Hastie et al. (2001). The data is available on the web at http://www-stat.stanford.edu/~tibs/ElemStatLearn. The data consists of log-periodograms of five different phonemes repeatedly pronounced by 50 males. The length of each periodogram is 256. Here we will focus on the phonemes 'aa' and 'ao', whose discrimination is not trivial because they sound alike. There is a total of 695 repetitions of the phoneme 'aa' (519 in the training set and 176 in the test set) and 1022 of the phoneme 'ao' (759 in the training set and 263 in the test set). We first estimate β on the training data, and then classify the test data. As a measure of performance we use the total misclassification error (MCE), which is the number of misclassified test observations.

The following methods were used:

- LD: The classical (non-penalized) logistic discrimination.
- LDQC: Logistic discrimination with quadratic penalization of the coefficients (the maximizer of (5)).
- LDQD: Logistic discrimination with quadratic penalization of the differences between consecutive coefficients (the maximizer of (6)).

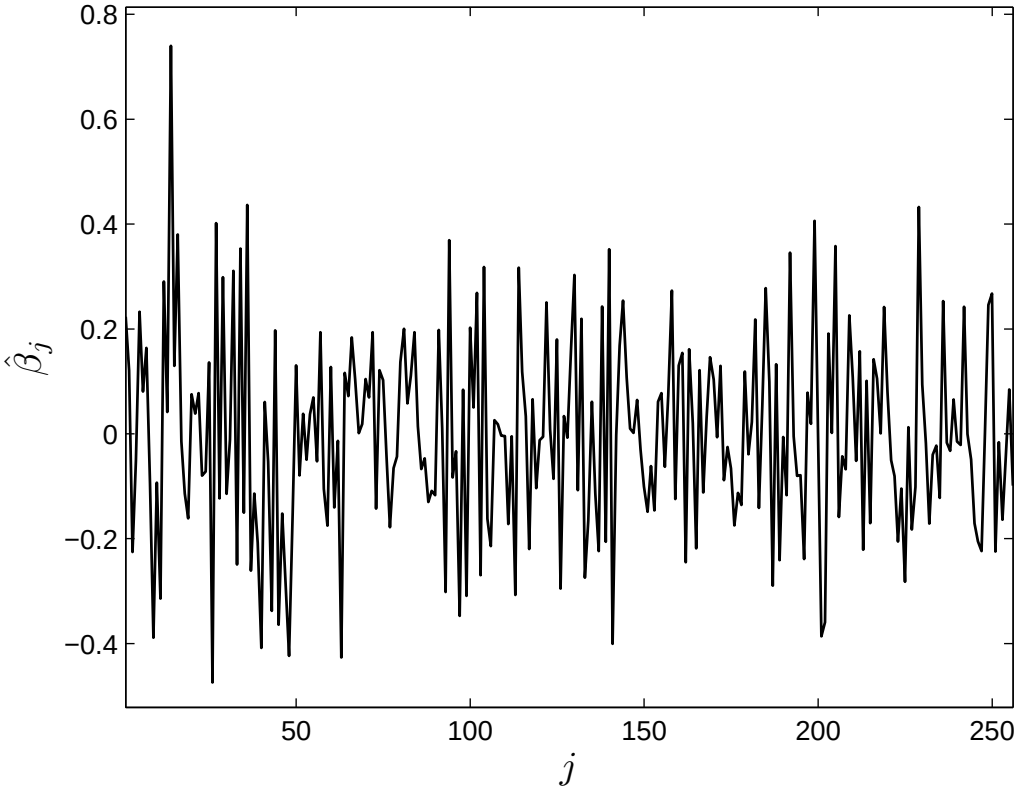

Figure 2: Phoneme recognition example. LD estimators.

- LDTV: Logistic discrimination with total-variation penalization (the maximizer of (7)).

The estimators LD, LDQC and LDQD were computed with Newton-Raphson algorithms, while the LDTV estimator was computed with the algorithm described in Section 3. The optimal penalization parameter λ was chosen by four-fold cross-validation. We call this estimator 'CV opt' in Table 1. Since part of the MCE of the estimators is due to the cross-validation selection of λ, we include 'true opt' in Table 1, which is the λ that minimizes the MCE on the test set. This is, effectively, the lowest attainable MCE for a given family of estimators, allowing us to judge the performance of an estimator independently of the method used to select λ.

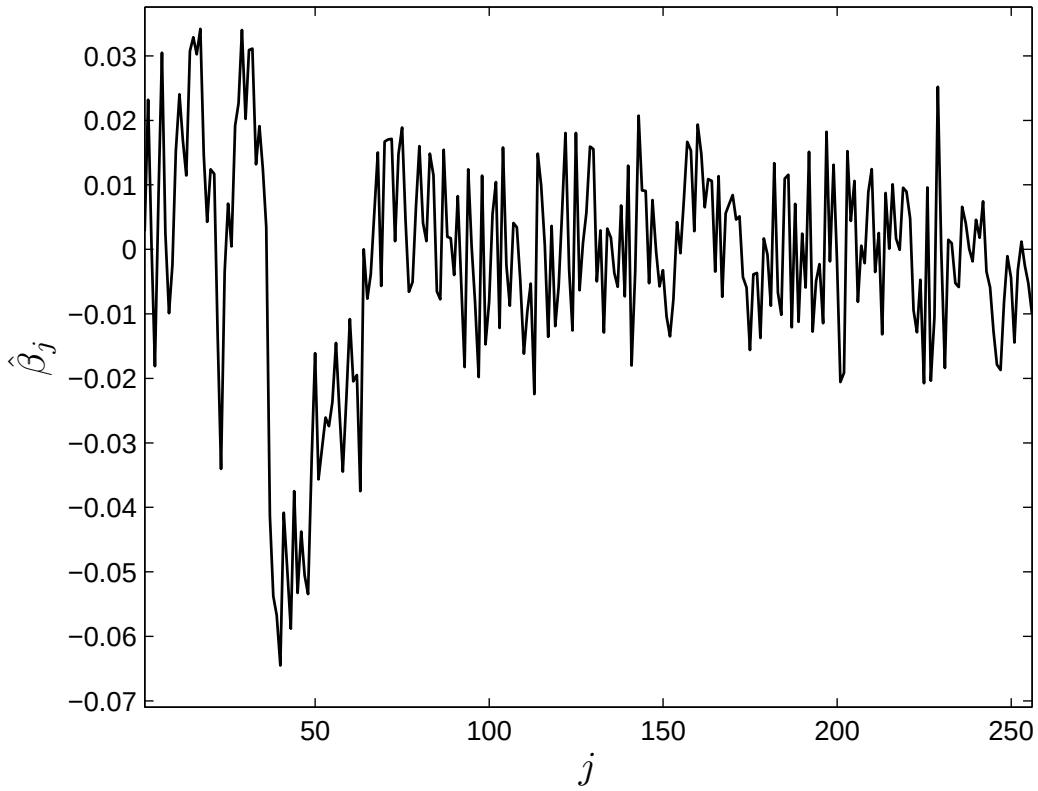

Figure 3: Phoneme recognition example. LDQC estimators.

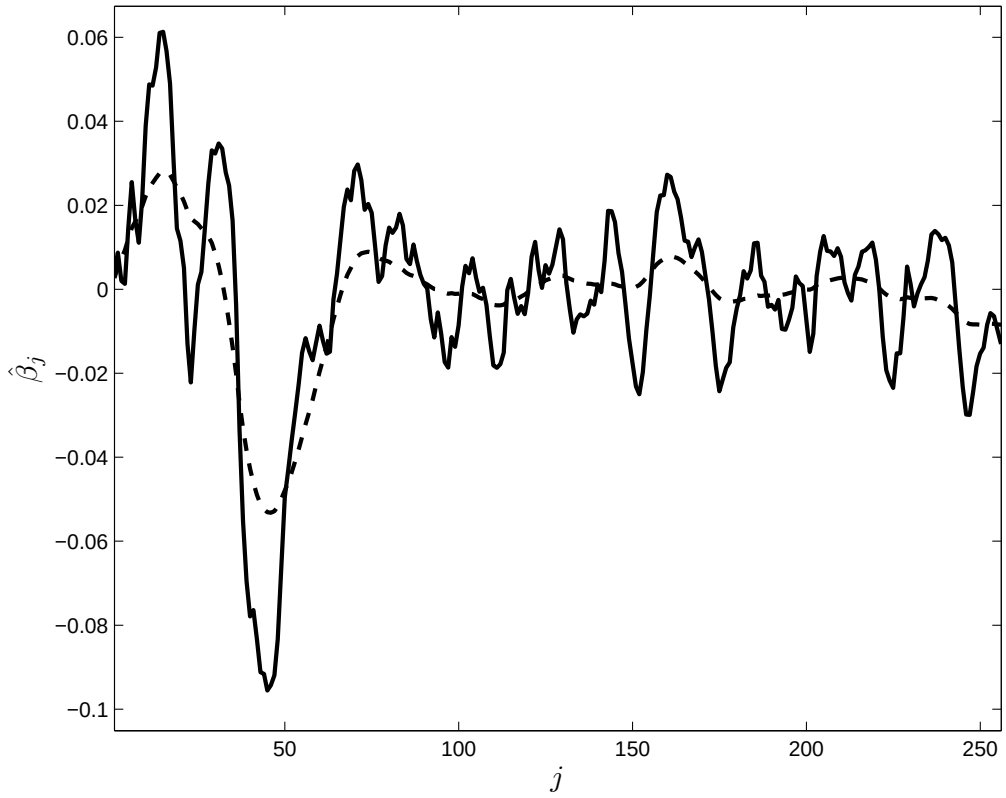

Figure 4: Phoneme recognition example. LDQD estimators: cross-validated optimum (solid line) and true optimum (dashed line).

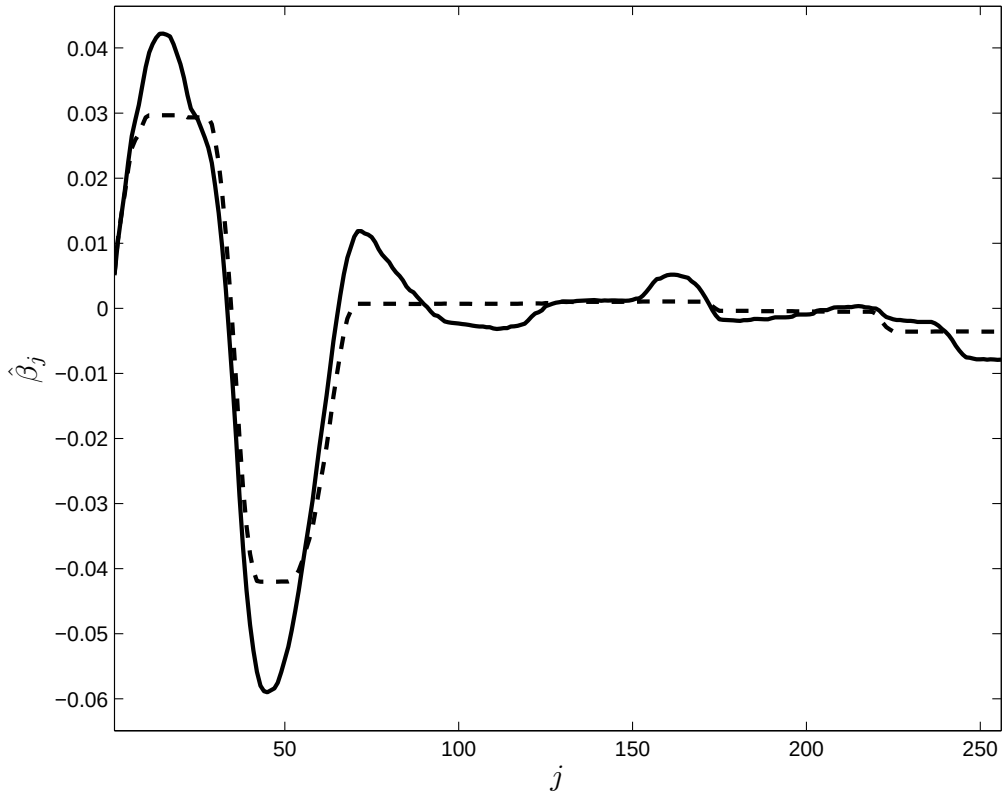

Figure 5: Phoneme recognition example. LDTV estimators: cross-validated optimum (solid line) and true optimum (dashed line).

It is clear from the misclassification errors in Table 1 that classical logistic discrimination does a poor job. The MCE on the training set is the smallest of the four estimators, but the MCE on the test set is the largest. Additionally, the estimated coefficients (Fig. 2) are so irregular that they are impossible to interpret; no particular range of frequencies seem to have more discriminatory power than others.

The regularized estimator LDQC (for which 'CV opt' and 'true opt' coincide) is a clear improvement, showing a smaller MCE on the test set and a more discernible behavior of $\hat{\boldsymbol{\beta}}$ (Fig. 3). We now see that low frequencies have more discriminatory power, as expected from Fig. 1, although the estimator is still very irregular.

The other quadratically penalized estimator, LDQD, does not represent an improvement from the MCE point of view, but the estimated coefficients do look better. Figure 4 shows both the cross-validated optimum ($\lambda = 2$) and the true optimum ($\lambda = 32$). The latter is indeed a very smooth function and, as expected, is practically zero for $j \geq 100$.

The total-variation regularized estimators (Fig. 5), both the cross-validated optimum ($\lambda = 1/32$) and the true optimum ($\lambda = 1/4$), show less variability than the previous estimators. The MCE attained by cross-validation is only marginally larger than those of LDQC and LDQD (18.9% as opposed to 18.7%), but the true optimum outperforms both at 18.2%, yielding the best MCE for this data. Note that the true optimum is flat and practically zero for high frequencies, confirming our intuition that most of the discriminatory power is due to the low frequencies (x_j with $1 \leq j \leq 70$). In conclusion, we can say that total-variation regularization outperforms the other methods for this data set, although there is room for improvement in the selection of the penalization parameter.

5 Simulations

The example in Section 4 allowed us to compare the MCEs of different methods and, to a lesser extent, the quality of estimation of $\boldsymbol{\beta}$. But the latter is hard to assess

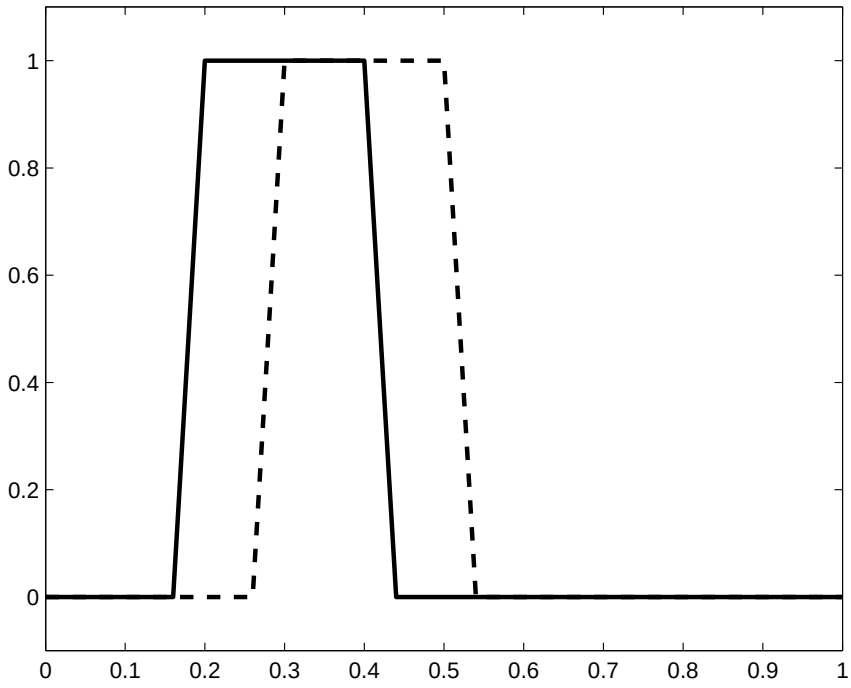

Figure 6: Simulations. Means for groups 1 (solid line) and 2 (dashed line).

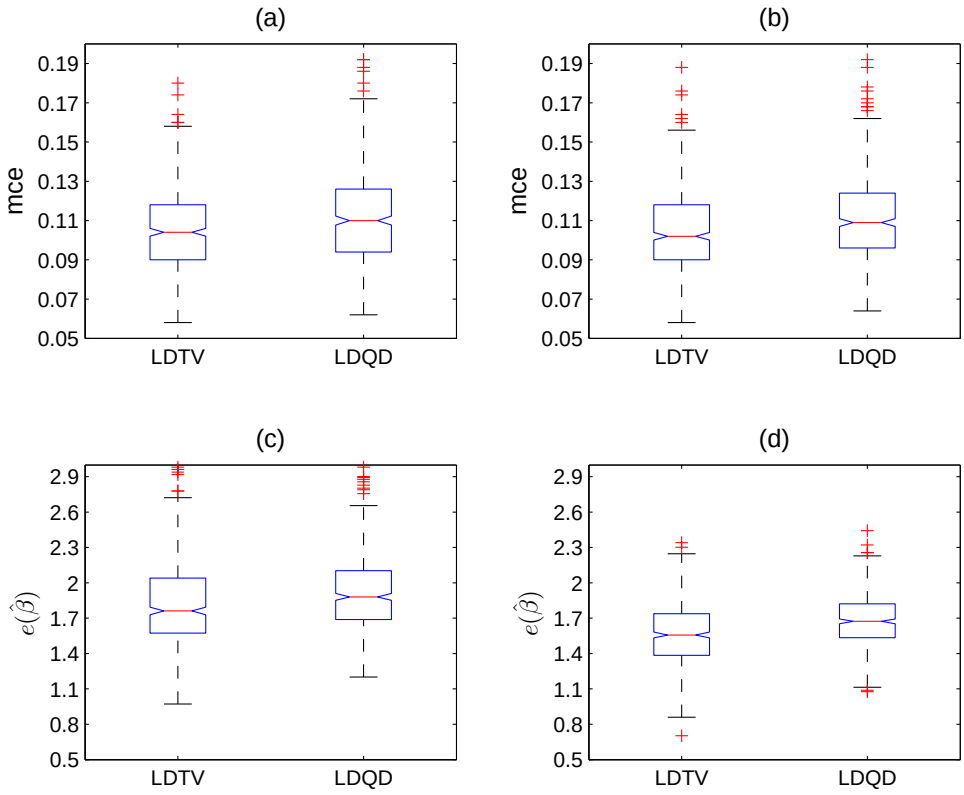

Figure 7: Simulations. Results for $n_{\text{train}} = 50$, $s = 1$. Misclassification errors for cross-validated estimators (a) and oracle estimators (b), and estimation errors for cross-validated estimators (c) and oracle estimators (d).

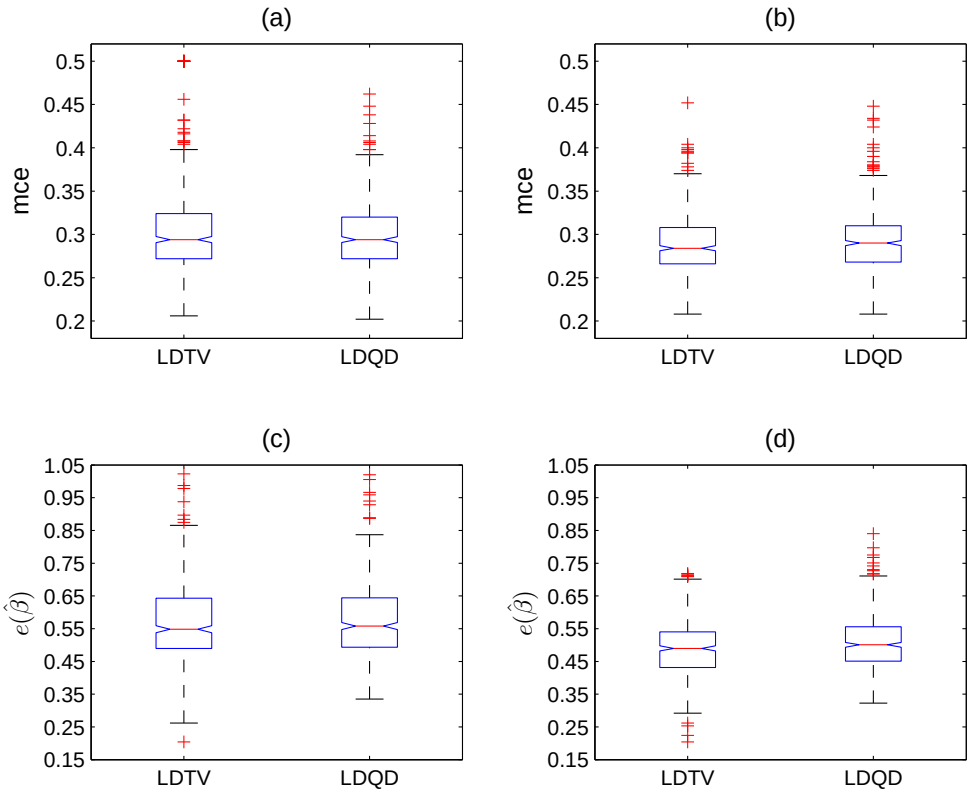

Figure 8: Simulations. Results for $n_{\text{train}} = 50$, $s = 2$. Misclassification errors for cross-validated estimators (a) and oracle estimators (b), and estimation errors for cross-validated estimators (c) and oracle estimators (d).

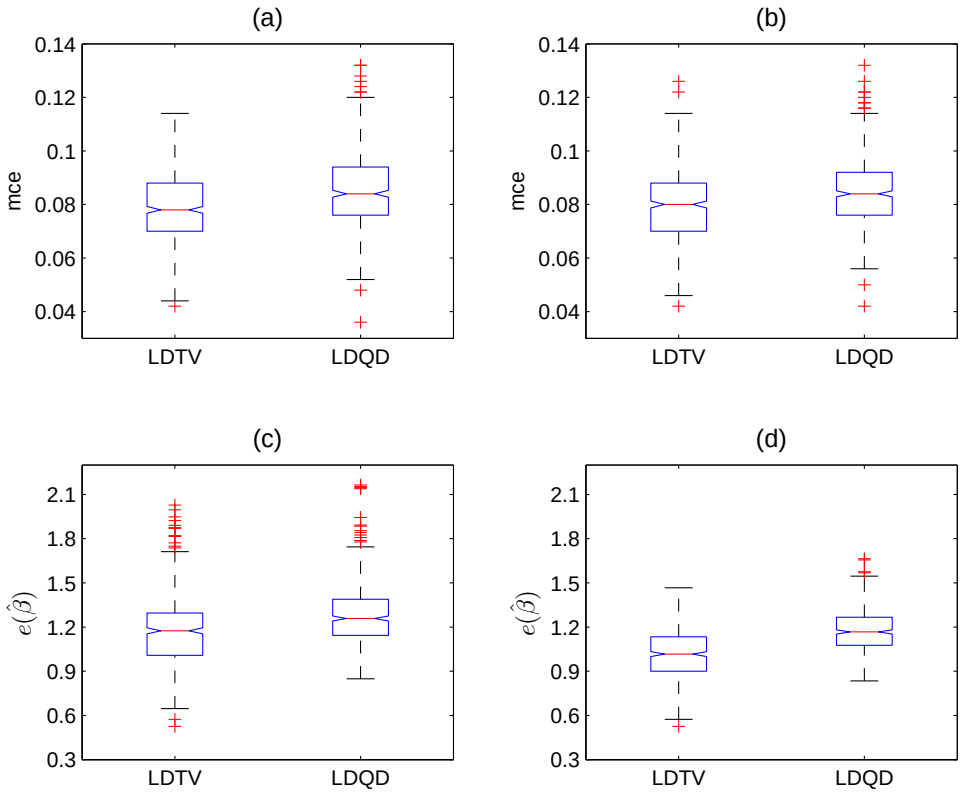

Figure 9: Simulations. Results for $n_{\text{train}} = 200$, $s = 1$. Misclassification errors for cross-validated estimators (a) and oracle estimators (b), and estimation errors for cross-validated estimators (c) and oracle estimators (d).

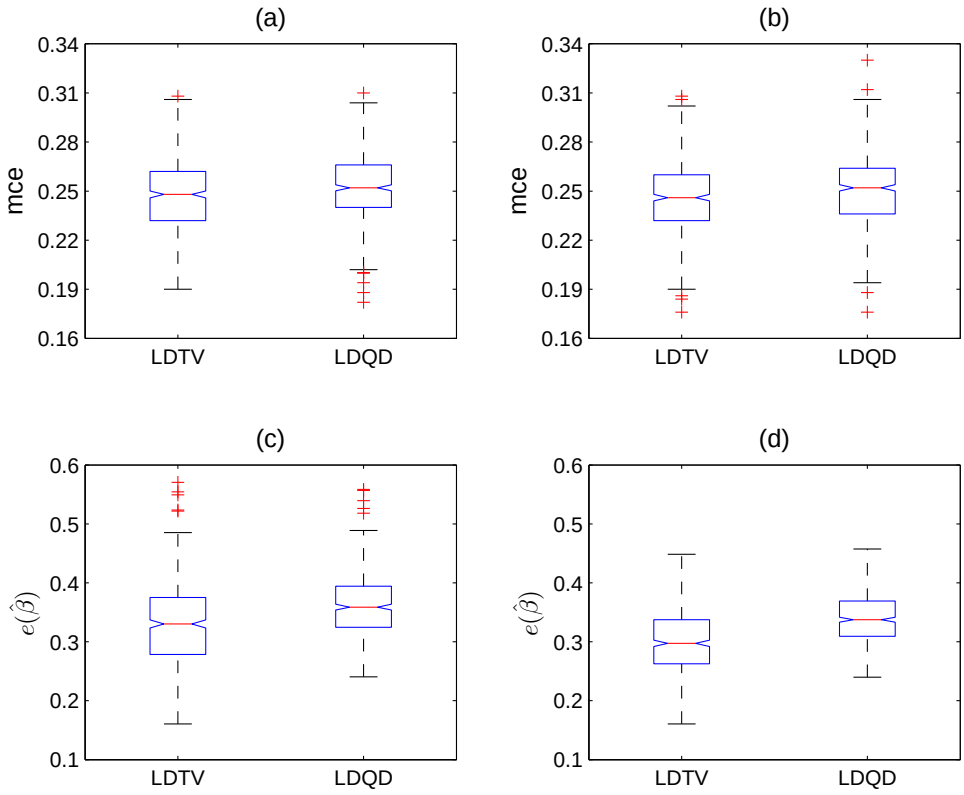

Figure 10: Simulations. Results for $n_{\text{train}} = 200$, $s = 2$. Misclassification errors for cross-validated estimators (a) and oracle estimators (b), and estimation errors for cross-validated estimators (c) and oracle estimators (d).

from a real-life dataset, for which there are no known 'true' parameters. So we ran some simulations that resemble the phoneme recognition problem (in that the $\boldsymbol{x}_i$s are discretized stochastic processes and the two group means differ only within a narrow range of values) but for which $\boldsymbol{\beta}$ is known.

We generated samples from two 51-dimensional multivariate Normal distributions with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, and common covariance matrix $\Sigma = s^2\mathbb{I}$. The mean vector $\boldsymbol{\mu}_1$ was computed by linear interpolation of the points $(0, 0)$, $(0.16, 0)$, $(0.2, 1)$, $(0.4, 1)$, $(0.44, 0)$ and $(1, 0)$, while $\boldsymbol{\mu}_2$ was computed by linear interpolation of the points $(0, 0)$, $(0.26, 0)$, $(0.3, 1)$, $(0.5, 1)$, $(0.54, 0)$ and $(1, 0)$; see Fig. 6. The same number of training and test observations were drawn from each distribution (so the group probabilities are both 0.5).

For two Normal populations with common covariance and equal group probabilities, the optimal discrimination rule is linear and (1) holds with $\alpha = -0.5(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ and $\boldsymbol{\beta} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$. The corresponding misclassification error, called optimum error rate (OER), is $\Phi(-\Delta/2)$ with $\Delta = \{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\}^{1/2}$ (see Johnson and Wichern (2002)).

Four scenarios were simulated (with 500 replications each): the combinations of the sample sizes $n_{\text{train}} = 50$ and $n_{\text{train}} = 200$ with the standard deviations $s = 1$ and $s = 2$. The MCE was computed on test sets of 500 observations. We considered two estimators: logistic discrimination with total-variation penalization (LDTV) and with quadratic penalization of the coefficient differences (LDQD). The penalization parameter λ was chosen by four-fold cross-validation, as in Section 4. We also computed the 'oracle' estimators, corresponding to the λ that minimizes the estimation error $e(\hat{\boldsymbol{\beta}}) = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|$. Comparing cross-validated with oracle estimators gives a better idea of how much of the error is due to cross-validation and how much is due to the estimation method itself.

The results are shown in Figs. 7–10 and Tables 2 and 3. We see that the total-variation regularization outperforms LDQD in all situations but one. Only in the most difficult scenario (small sample size, $n_{\text{train}} = 50$, and high noise, $s = 2$) is LDTV outperformed by LDQD, and even in that case, only for the λ chosen by cross-validation, not for the 'oracle' estimator. So we can attribute this exception

to the tendency of cross-validation to select small penalization parameters. For the other cases we see that total-variation regularization provides better estimators of the coefficients and smaller misclassification errors than quadratic regularization.

Table 2: Simulations: Mean MCE (and standard errors).

		CV estimators		Oracle estimators	
Model	OER	LDTV	LDQD	LDTV	LDQD
$n_{\text{train}} = 50$, $s = 1$	0.067	0.105 (<0.001)	0.112 (0.001)	0.104 (<0.001)	0.111 (0.001)
$n_{\text{train}} = 50$, $s = 2$	0.227	0.314 (0.003)	0.299 (0.002)	0.288 (0.002)	0.292 (0.002)
$n_{\text{train}} = 200$, $s = 1$	0.067	0.078 (<0.001)	0.085 (<0.001)	0.080 (<0.001)	0.084 (<0.001)
$n_{\text{train}} = 200$, $s = 2$	0.227	0.247 (0.001)	0.252 (0.001)	0.246 (0.001)	0.251 (0.001)

Table 3: Simulations: Mean estimation errors of $\hat{\boldsymbol{\beta}}$ (and standard errors).

	CV estimators		Oracle estimators	
Model	LDTV	LDQD	LDTV	LDQD
$n_{\text{train}} = 50$, $s = 1$	2.032 (0.043)	2.073 (0.034)	1.556 (0.012)	1.684 (0.010)
$n_{\text{train}} = 50$, $s = 2$	0.575 (0.006)	0.587 (0.008)	0.488 (0.004)	0.507 (0.004)
$n_{\text{train}} = 200$, $s = 1$	1.178 (0.015)	1.295 (0.011)	1.017 (0.008)	1.173 (0.006)
$n_{\text{train}} = 200$, $s = 2$	0.329 (0.003)	0.362 (0.002)	0.300 (0.002)	0.339 (0.002)

References

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96** 1348–1360.

Fan, J. and Peng, H. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *Ann. Statist.* **32** 928–961.

Hastie, T. and Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer, New York.

Johnson, R. A. and Wichern, D.W. (2002) *Applied Multivariate Statistical Analysis.* Prentice Hall Upper Saddle River, New Jersey.

Krzanowski, W.J. (1977). The performance of Fisher's linear discriminant function under non-optimal conditions. *Technometrics* **19** 191–200.

Nocedal, J. and Wright, S.J. (1999). *Numerical Optimization.* Springer, New York.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *J. R. Statist. Soc. Ser. B* **58** 267–288.

Zhu, J. and Hastie, T. (2004). Classification of gene microarrays by penalized logistic regression. *Biostat.* **5** 427–443.

DEPARTMENT OF MATHEMATICAL SCIENCES
UNIVERSITY OF WISCONSIN–MILWAUKEE
3200 N. CRAMER ST., EMS E487
MILWAUKEE, WI 53211, USA
E-MAIL: gervini@uwm.edu